

# Session 12: Closing

## Softwaretechnologie: Java II

Nils Reiter

`nils.reiter@uni-koeln.de`

July 13, 2022

# Section 1

## Recap

# Unit-Testing

- ▶ Documentation of expected output/behaviour of functions/methods
- ▶ Automatic verification that code works as expected
- ▶ Junit: Assert functions
- ▶ Integration in Maven and Eclipse
- ▶ Exercise 13

## Und wie geht's weiter?

- ▶ Programmieren ist *die* am breitesten anwendbare Technik, die Sie (hier) lernen
- ▶ Nur durch Training wird man besser/schneller im Programmieren

## Und wie geht's weiter?

- ▶ Programmieren ist *die* am breitesten anwendbare Technik, die Sie (hier) lernen
- ▶ Nur durch Training wird man besser/schneller im Programmieren

### Ideen zum Dranbleiben

- ▶ Suchen Sie sich ein kleines Projekt aus ihrem Leben
  - ▶ Beispiele: Spiele, WG-Putzpläne/-Kasse, Analyse von (offenen) Daten, ...
- ▶ Schauen Sie sich eine andere Programmiersprache an
  - ▶ Hint: Die allermeisten Konzepte in Java gibt es auch in C++, Python, Javascript, Perl, ...
- ▶ Schauen Sie sich eine außergewöhnliche/"exotische" Programmiersprache an
  - ▶ Lisp, Prolog, Assembler
- ▶ Java gibt es in verschiedenen Versionen
  - ▶ Versuchen Sie zu recherchieren und zu verstehen, was jeweils neu war

## Section 2

# Modulprüfung

# Formalitäten

- ▶ Größeres Programmierprojekt nach Spezifikation
- ▶ Vorlage ist als ZIP-Datei in [Ilias](#) bereitgestellt
- ▶ Abgabe
  - ▶ Ausschließlich als ZIP-Datei per Ilias
  - ▶ Bis 15.09.2022, 23:55
- ▶ Gruppen bis maximal drei Personen möglich
  - ▶ bei klarer Kennzeichnung wer was gemacht hat
  - ▶ Individuelle Code-Dokumentation und Abgabe
- ▶ Identische Abgaben werden als Täuschungsversuch gewertet

demo



# Die Aufgabe

1. Implementierung und Bestehen aller Unit-Tests
  2. Interne Dokumentation
- 
3. Suchfunktion unterstützt Wildcards
  4. Sortier-Funktion für abonnierte Feeds

Oberhalb der Linie: Mindestanforderung zum Bestehen

# Tipps

- ▶ Die Vorlage enthält eine **ausführliche Anleitung**. Lesen Sie sie sorgfältig.
- ▶ Machen Sie einen Plan: Was ist der Kern des Problems und wie können Sie es lösen?
- ▶ Nehmen Sie sich Zeit
  - ▶ Die Modulprüfung ist umfangreicher als die wöchentlichen Aufgaben.
- ▶ Verwenden Sie gut zur Versionierung
- ▶ Gruppen:
  - ▶ Sprechen Sie sich regelmäßig und häufig ab.
  - ▶ Treffen Sie klare (und verbindliche) Absprachen.

## Bei Fragen

- ▶ Kontaktieren Sie uns gerne!  
(aber nicht am 15.09., 23:00)